

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-358280

(43)Date of publication of application : 13.12.2002

(51)Int.Cl.

G06F 15/00

(21)Application number : 2001-185207

(71)Applicant : RICOH CO LTD

(22)Date of filing : 31.05.2001

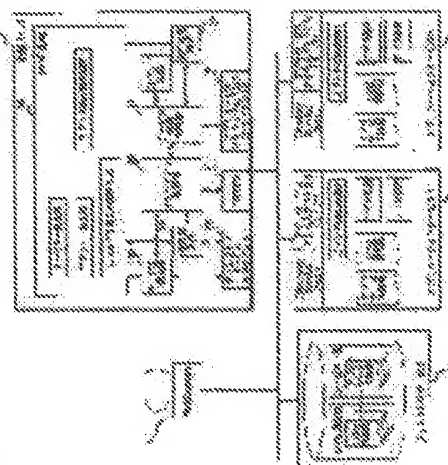
(72)Inventor : IMAI TATSUYA

## (54) CLIENT SERVER SYSTEM

## (57)Abstract:

PROBLEM TO BE SOLVED: To make it possible to use a new application service from a client terminal device without adding a new module to a relay server when the new application service is added on a network.

SOLUTION: In a client server system wherein the client terminal device 1, an application service device 2 having an application service means and the relay server 3 are connected through the network, and the relay server 3, depending on a request from the client terminal device 1, makes a request of the application service means for processing, and receives the processing result from the application service means to send it back to the client terminal device 1, the system is constituted such that the relay server 3 acquires the information on a processing request method regarding the application service designated from the client terminal device 1, and creates HTML data based on the information to send the data back to the client terminal device 1.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-358280

(P2002-358280A)

(43) 公開日 平成14年12月13日 (2002.12.13)

(51) Int.Cl.<sup>7</sup>

G 0 6 F 15/00

識別記号

3 1 0

F I

G 0 6 F 15/00

特 許 庁 (参考)

3 1 0 B 5 B 0 8 5

審査請求 未請求 請求項の数10 O L (全 13 頁)

(21) 出願番号 特開2001-185207(P2001-185207)

(22) 出願日 平成13年5月31日 (2001.5.31)

(71) 出願人 000000747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72) 発明者 今井 謙也

東京都大田区中馬込1丁目3番6号 株式

会社リコー内

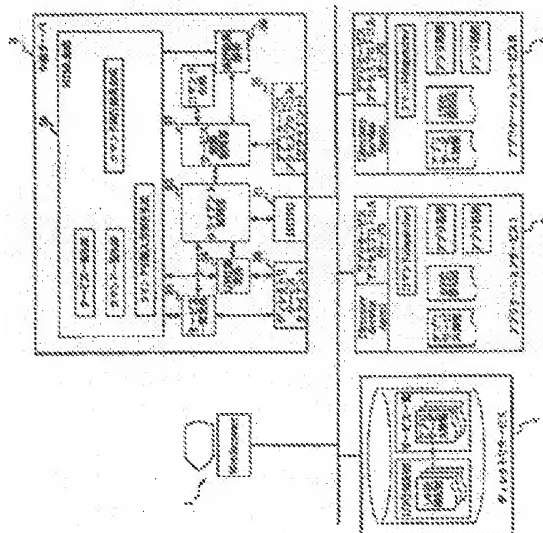
Fターム(参考) 5B085 M01 B032 B037

(54) 【発明の名称】 クライアントサーバシステム

(57) 【要約】

【課題】 ネットワーク上に新たなアプリケーションサービスが追加されたとき、中継サーバに新たなモジュールを追加することなくクライアント端末装置から新しいアプリケーションサービスを利用できるようにする。

【解決手段】 クライアント端末装置1、アプリケーションサービス手段を備えたアプリケーションサーバ装置2、中継サーバ3がネットワークを介して接続され、その中継サーバ3がクライアント端末装置1からの要求によりアプリケーションサービス手段に処理を依頼し、処理結果をアプリケーションサービス手段から受信してクライアント端末装置1へ返送するクライアントサーバシステムにおいて、中継サーバ3が、クライアント端末装置1から指定されたアプリケーションサービスについて処理依頼方法に関する情報を取得し、その情報に基づいてURLデータを生成し、クライアント端末装置1へ返送する構成にした。



## 【特許請求の範囲】

【請求項1】 ブラウザを搭載したクライアント 端末装置と、そのクライアント 端末装置からの要求により処理を行う一つ以上のアプリケーションサービス手段と、  
 HTTPサーバを含む中継サーバとがネットワークを介して接続され、その中継サーバがクライアント 端末装置からの要求情報を受信し、その要求情報に基づいてアプリケーションサービス手段に処理を依頼し、処理結果をそのアプリケーションサービス手段から受信してクライアント 10  
 端末装置へ返送するクライアント サーバシステムにおいて、中継サーバが、クライアント 端末装置からの要求情報中に指定されたアプリケーションサービスについてサービスに対する処理依頼方法に関する情報を取得し、取得した情報に基づいてHTMLデータを生成し、クライアント 端末装置へ返送することを特徴とするクライアント サーバシステム。

【請求項2】 請求項1記載のクライアント サーバシステムにおいて、クライアント 端末装置で前記要求情報を作成するために、中継サーバの生成するHTMLデータが各アプリケーションサービスの各コマンドに対応した入力 20  
 項目を備え、アプリケーションサービスがクライアント 端末装置からの要求により複数の異なる処理を実行可能で、クライアント 端末装置からの処理要求が処理内容に対応したコマンドを送信する形で行われる場合、前記入力項目を用いてクライアント 端末装置のブラウザ上で入力または選択された内容の中継サーバへ送ることを特徴とするクライアント サーバシステム。

【請求項3】 請求項2記載のクライアント サーバシステムにおいて、アプリケーションサービス手段が各コマンドの実行にコマンドの指定の他にパラメータの指定が 30  
 必要な場合、中継サーバが、前記ブラウザ上におけるパラメータ値の指定を可能にさせる入力項目をもつHTMLデータを生成することを特徴とするクライアント サーバシステム。

【請求項4】 請求項1、請求項2、または請求項3記載のクライアント サーバシステムにおいて、ネットワーク上の全てのアプリケーションサービスを管理するディレクトリサービス手段をネットワーク内に備え、アプリケーションサービスに対する処理の依頼方法に関する情報を前記ディレクトリサービス手段から取得する構成にしたことを特徴とするクライアント サーバシステム。 40

【請求項5】 請求項1、請求項2、または請求項3記載のクライアント サーバシステムにおいて、アプリケーションサービスに対する処理の依頼方法に関する情報を各アプリケーションサービス手段が管理し、各アプリケーションサービス間で共通な特定のコマンドを各アプリケーションサービス手段へ送ることにより前記依頼方法に関する情報を取得する構成にしたことを特徴とするクライアント サーバシステム。

【請求項6】 請求項1乃至請求項5のいずれかに記載 50

のクライアント サーバシステムにおいて、クライアント 端末装置上で、送られてきた入力項目を含むHTMLデータをブラウザ上に表示し、その入力項目に入力値を入力させるか入力値を選択させるかして指定させ、その入力値をクライアント 端末装置から中継サーバへ送り、アプリケーションサービスに対する処理の依頼方法に関する情報にしたがって中継サーバが前記入力値からサービスに対する実際のコマンドおよびパラメータを生成し、アプリケーションサービス手段へ送信することを特徴とするクライアント サーバシステム。

【請求項7】 請求項6記載のクライアント サーバシステムにおいて、中継サーバが、コマンド送信に対する応答としてアプリケーションサービス手段から処理結果を受信し、その処理結果を元にHTMLデータを生成し、生成したHTMLデータをクライアント 端末装置へ返送することを特徴とするクライアント サーバシステム。

【請求項8】 請求項7記載のクライアント サーバシステムにおいて、アプリケーションサービス手段へのコマンド送信に対する処理結果としてアプリケーションサービス手段から当該アプリケーションサービスまたは別のアプリケーションサービスのサービス識別子が返送され、中継サーバが返送されたサービス識別子に対応するアプリケーションサービスに関する処理の依頼方法を取得し、取得した情報に基づきHTMLデータを生成し、生成したHTMLデータをクライアント 端末装置へ返送することを特徴とするクライアント サーバシステム。

【請求項9】 請求項1、請求項2、または請求項3記載のクライアント サーバシステムにおいて、アプリケーションサービスを、サービスオブジェクトとして外部インターフェースから参照可能なドメインウェアを用いて実装し、アプリケーションサービスに対する処理の依頼方法に関する情報として前記サービスオブジェクトの外部インターフェースに関する情報をアプリケーションサービス手段から取得する構成にしたことを特徴とするクライアント サーバシステム。

【請求項10】 請求項9記載のクライアント サーバシステムにおいて、前記ドメインウェアとしてJAVA(登録商標)言語のHTML機能を用い、JSP言語のpagelet機能を用いてサービスオブジェクトの外部インターフェースに関する情報を取得することを特徴とするクライアント サーバシステム。

## 【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ネットワークを介してアプリケーションサービスを行なうことができるクライアント サーバシステムに関する。

【0002】

【従来の技術】一般的に言って、クライアント サーバシステムとは、所定のアプリケーション処理を行うネットワーク上のアプリケーションサービス装置などアプリケ

アプリケーションサービス手段に対してクライアント 端末装置が処理を依頼し、その結果を取得するシステムである。このようなシステムでは、アプリケーションサービス手段に対するアクセスを行い、また、アプリケーションサービス手段からの処理結果を受信解析するためのライブラリがクライアント 端末装置側に必要で、通常のこのようなライブラリはアプリケーションサービス毎に用意される。一般には複数のアプリケーションサービスに対してクライアント 端末装置がアクセスする環境ではクライアント 端末装置とアプリケーションサービス手段上のプログラム間でのデータ交換プロトコルを統一するために分散処理用ミドルウェアが用いられることが多い。このようなミドルウェアの例としてはCGIのUNIX、Microsoft社ASP/JSP、CGIのPHPなどが挙げられる。しかし、通常のこれらのミドルウェアはデータ交換のプロトコルは統一するが、アプリケーションサービスの提供する機能内容、その機能の依頼方法、依頼結果として返送されるデータの解析方法等に関する情報をクライアント 端末装置側が取得することを可能としていない。このため、クライアント 端末装置側ではやはりアプリケーションサービスの提供する機能毎に設計されたライブラリを用意し、それをクライアント 端末装置側に導入する必要がある。

【0003】ところで、近年のWeb技術の普及はほぼ全てのパーソナルコンピュータにWebブラウザを搭載させた。そこで、このWebブラウザを搭載したクライアント 端末装置とアプリケーションサービス手段との間にHTTPサーバを配置し、HTTPサーバでクライアント 端末装置からの指示を受信し、それに基づいてHTTPサーバがネットワーク上のアプリケーションサービス手段に対して処理を依頼し、その処理結果をHTMLデータに変換してWebブラウザに返送するという3層構造をもったシステムが多く提案されている。このようなシステムでは、クライアント 端末装置上にはWebブラウザのみが必要で利用者はミドルウェアなど特殊なライブラリをクライアント 端末装置に導入する必要から開放される。しかし、このようなシステムでも、ネットワーク上のアプリケーションサービス手段の提供する機能およびそこへのアクセス方法はアプリケーションサービス毎に異なるため、HTTPサーバがネットワーク上にあるアプリケーションサービス手段へアクセスし、その結果をHTMLデータに変換するという処理のためには個々のアプリケーションサービスに依存したライブラリがHTTPサーバ上に導入されていることが必要になる。そのため、ネットワーク上に新しいアプリケーションサービスが追加された場合には、そのアプリケーションサービスを利用するための処理をHTTPサーバに追加する必要がある。

【0004】

【発明が解決しようとする課題】しかしながら、通常、Webブラウザ上に表示される内容とネットワーク上のアプリケーションサービス手段が提供する機能との間の

一定の変換規則がないため、従来技術においては、新しいアプリケーションサービスが導入されるたびに、それをWebブラウザから利用するためのHTMLデータのデザインを決定し、HTMLデータとアプリケーションサービスの提供する機能との間での変換方式をアプリケーションサービス毎に開発する必要があった。また、ネットワーク上に各種アプリケーションサービスが一時的に追加されたり、切断されたりするような環境には対応ができなかった。

【0005】本発明の目的は、このような従来技術の問題を解決することにある。具体的には、クライアントサーバシステムにおいて、クライアント 端末装置とアプリケーションサービス手段の間に中継サーバを配置し、この中継サーバがクライアント 端末装置からの要求に応じてネットワーク上のアプリケーションサービス手段にアクセスする場合に、ネットワーク上に新たなアプリケーションサービスが追加されたときに、中継サーバに新たなモジュールを追加することなく、クライアント 端末装置からこの新しいアプリケーションサービスを利用することを可能とすることにある。また、アプリケーションサービス手段への処理依頼がアプリケーションサービス手段へコマンドを送信する形で実行可能なシステムにおいて、アプリケーションサービスの各コマンドを利用者が直接指定して実行することが可能なユーザインターフェースを実現することにある。また、アプリケーションサービス手段への処理依頼がコマンドに加え複数のパラメータを送信する形で実行可能なシステムにおいて、コマンドに対し必要なパラメータを利用者が直接指定して実行することが可能なユーザインターフェースを実現することにある。また、アプリケーションサービス手段への処理依頼がアプリケーションサービス手段へコマンドを送信する形で実行可能なシステムにおいて、アプリケーションサービスの処理可能なコマンドの一覧を中継サーバに予め登録しておかなくても、新たなサービスが追加された時点で、中継サーバがそれらの情報を動的に収集可能なシステムを実現することにある。また、アプリケーションサービス手段による処理結果がそれと連携する別のアプリケーションサービスの処理と関連する場合、この別のアプリケーションサービス手段へのアクセスを利用者に提供することにある。また、中継サーバが新たなアプリケーションサービスが追加された時点でそれらの情報を動的に収集可能にすることにある。

【0006】

【課題を解決するための手段】前記の課題を解決するために、請求項1記載の発明では、ブラウザを搭載したクライアント 端末装置と、そのクライアント 端末装置からの要求により処理を行う一つ以上のアプリケーションサービス手段と、HTTPサーバを含む中継サーバとがネットワークを介して接続され、その中継サーバがクライアント 端末装置からの要求情報を受信し、その要求情報に基

5

ついてアプリケーションサービス手段に処理を依頼し、処理結果をそのアプリケーションサービス手段から受信してクライアント端末装置へ返送するクライアントサーバシステムにおいて、中継サーバが、クライアント端末装置からの要求情報中に指定されたアプリケーションサービスについてサービスに対する処理依頼方法に関する情報を取得し、取得した情報に基づいてXMLデータを生成し、クライアント端末装置へ返送することを特徴とする。また、請求項2記載の発明では、請求項1記載の発明において、クライアント端末装置で前記要求情報を作成するために、中継サーバの生成するXMLデータが各アプリケーションサービスの各コマンドに対応した入力項目を備え、アプリケーションサービスがクライアント端末装置からの要求により複数の異なる処理を実行可能で、クライアント端末装置からの処理要求が処理内容に対応したコマンドを送信する形で行われる場合、前記入力項目を用いてクライアント端末装置のブラウザ上で入力または選択された内容の中継サーバへ送ることを特徴とする。また、請求項3記載の発明では、請求項2記載の発明において、アプリケーションサービス手段が各コマンドの実行にコマンドの指定の他にパラメータの指定が必要な場合、中継サーバが、前記ブラウザ上におけるパラメータ値の指定を可能にさせる入力項目をもつXMLデータを生成する構成にした。また、請求項4記載の発明では、請求項1、請求項2、または請求項3記載の発明において、ネットワーク上の全てのアプリケーションサービスを管理するディレクトリサービス手段をネットワーク内に備え、アプリケーションサービスに対する処理の依頼方法に関する情報を前記ディレクトリサービス手段から取得することを特徴とする。

【0007】また、請求項5記載の発明では、請求項1、請求項2、または請求項3記載の発明において、アプリケーションサービスに対する処理の依頼方法に関する情報を各アプリケーションサービス手段が管理し、各アプリケーションサービス間で共通な特定のコマンドを各アプリケーションサービス手段へ送ることにより前記依頼方法に関する情報を取得することを特徴とする。また、請求項6記載の発明では、請求項1乃至請求項5のいずれかに記載の発明において、クライアント端末装置上で、送られてきた入力項目を含むXMLデータをブラウザ上に表示し、その入力項目に入力値を入力させるか入力値を選択させるかして指定させ、その入力値をクライアント端末装置から中継サーバへ送り、アプリケーションサービスに対する処理の依頼方法に関する情報にしたがって中継サーバが前記入力値からサービスに対する実際のコマンドおよびパラメータを生成し、アプリケーションサービス手段へ送信することを特徴とする。また、請求項7記載の発明では、請求項8記載の発明において、中継サーバが、コマンド送信に対する応答としてアプリケーションサービス手段から処理結果を受信し、そ

6

の処理結果を元にXMLデータを生成し、生成したXMLデータをクライアント端末装置へ返送することを特徴とする。また、請求項8記載の発明では、請求項7記載の発明において、アプリケーションサービス手段へのコマンド送信に対する処理結果としてアプリケーションサービス手段から当該アプリケーションサービスまたは別のアプリケーションサービスのサービス識別子が返送され、中継サーバが返送されたサービス識別子に対応するアプリケーションサービスに対する処理の依頼方法を取得し、取得した情報に基づきXMLデータを生成し、生成したXMLデータをクライアント端末装置へ返送することを特徴とする。また、請求項9記載の発明では、請求項1、請求項2、または請求項3記載の発明において、アプリケーションサービスを、サービスオブジェクトとして外部インターフェースから参照可能なミドルウェアを用いて実装し、アプリケーションサービスに対する処理の依頼方法に関する情報として前記サービスオブジェクトの外部インターフェースに関する情報をアプリケーションサービス手段から取得することを特徴とする。また、請求項10記載の発明では、請求項9記載の発明において、前記ミドルウェアとしてJava言語のRMI/JRMI機能を用い、J2EE言語のReflection機能を用いてサービスオブジェクトの外部インターフェースに関する情報を取得することを特徴とする。

【0008】

【発明の実施の形態】以下、図面により本発明の実施の形態を詳細に説明する。図1は本発明の一実施例を示すクライアントサーバシステムのシステム構成図である。図示したように、この実施例のクライアントサーバシステムは、webブラウザを搭載した1台以上のクライアント端末装置1、それぞれ一つまたは複数のアプリケーションサービス手段を備えた一つ以上のアプリケーションサービス装置2、中継サーバ3、ディレクトリサービス装置4などから構成される。前記において、webブラウザを搭載したクライアント端末装置1と中継サーバ3の間の通信はHTTPを用いて行われる。このため、中継サーバ3にはHTTPサーバが搭載される。また、各アプリケーションサービス手段と中継サーバ3の間は分散オブジェクト環境を実現するミドルウェアで接続される。このようなミドルウェアの例としてはCORBA、J2EE、JMS、SOAP、XML、Single Object Access Protocol (SOAP)などがあげられる。このミドルウェア機能として、中継サーバ3にはアプリケーションサービス側にはアプリケーションサービスアクセスプロトコルサーバがそれぞれ搭載される。各アプリケーションサービス手段は外部からの処理要求をネットワーク経由で受信し、処理要求に従って処理を実行し、その処理結果を要求元へ返送する。各アプリケーションサービス手段への処理指示は処理依頼を指定するコマンドとその処理に対するパラメ



ークの形で構成される。アプリケーションサービス手段内部では受信したコマンドの種類に応じてコマンドディスパッチャ(dispatcher)により異なるアプリ処理手続きが起動される。また、クライアント端末装置1には、アプリケーションサービスの提供するコマンド一覧およびそのコマンドに対するパラメータの情報を取得する手段を備える。このような手段の実現方法としては、該当するアプリケーションサービス側へ送られたコマンドを送信することにより、アプリケーションサービスが記述されているインターフェース(I/F)仕様記述データ

10 を取得する方法、アプリケーションサービス側が内部の各アプリ処理手続きの持つAPI情報からI/F仕様記述データを動的に生成して送る方法がある。さらに、ネットワーク上に存在するディレクトリサービス装置4にアプリケーションサービス側のI/F仕様記述データを登録しておく方法も可能である。また、San Microsystems社のNMI/JMXなどのように自己参照機能(reflection)を提供するミドルウェアの場合、クライアント側モジュールであるproxyから対応するサーバのAPI仕様を取得することも可能である。

【0009】ディレクトリサービス装置4はネットワーク上で提供されているアプリケーションサービスの一覧情報を保存している。ディレクトリサービスの実装例としてはLight Weight Directory Protocol(LDAP)、Service Location Protocol(SLP)や、San Microsystems社のNMI/JMX機能などが挙げられる。SLPやJMXのように各アプリケーションサービスが自動的にその情報を登録する機能をもつ場合、名称、識別子、ネットワークアドレス等のサービスに関する基本情報からなるアプリサービス情報をアプリケーションサービスのDirectory Service 30 Agentが自動的にディレクトリサービス装置4に登録する。LDAPなどのように自動的な情報の登録機能を持たないディレクトリサービスでは管理者が各アプリケーションサービスのアプリサービス情報を手動で登録する。ディレクトリサービスへのアクセスはそのクライアント機能であるディレクトリサービスクライアント39を介して行われる。ネットワーク上にディレクトリサービスが存在しない構成では、クライアント端末装置上のディレクトリサービスクライアント機能が同報通信(multicastまたはbroadcast)を利用してネットワーク上の 40 全アプリケーションサービス手段のDirectory Service Agentにアクセスし、各Directory Service Agentが所属するアプリケーションサービスのアプリサービス情報を応答するという構成が可能である。このような実装を可能にするものとしては前述のService Location Protocol(SLP)の他に、Microsoft社のUniversal Plug&Play(UPnP)がある。中継サーバ3はWEBブラウザを搭載したクライアント端末装置1からの要求をHTTPで受信し、ネットワーク上のアプリケーションサービスの一覧とコマンド一覧をXMLデータで返送する。さらに、クライアント

端末装置1からの要求により特定のアプリケーションサービス手段へコマンドを送信し、そのコマンド実行結果をそのアプリケーションサービス手段から受信し、クライアント端末装置1へ返送する。

【0010】図1に示したように、中継サーバ3には、HTTPサーバ(HTTP)31を介して受信したクライアント端末装置1からの要求を処理するクライアント制御部32、ディレクトリサービス装置4へのアクセスその他の方法でネットワーク上のアプリケーションサービスの一覧を取得してXMLデータを生成させるサービス一覧取得部33、特定のアプリケーションサービスのI/F仕様記述データを取得してXMLデータとして生成するI/F仕様取得部34などがある。さらに、クライアント制御部32により起動され、アプリケーションサービスに対するコマンドを生成し、それをアプリサービスアクセスプロトコル(クライアント)35を介して実際のアプリケーションサービス手段へ送信し、次にそのコマンドの処理結果(実行結果)を受信し、その処理結果を実行結果解析部36を通してXMLデータを生成し、それをクライアント制御部32へ返送するサービス起動制御部37がある。また、実際にXMLデータを生成する手段として、サービス一覧取得部33から起動されるサービス一覧生成部、I/F仕様取得部34から起動されるコマンド一覧生成部とコマンド引数入力項目生成部、実行結果解析部36から起動されるコマンド実行結果生成部などから成るXML生成部38がある。

【0011】図2に、この実施例の処理の流れを示す。処理はクライアント端末装置1内のWEBブラウザが中継サーバ3にサービス一覧を要求することから開始する。WEBブラウザからの要求は中継サーバ3のHTTPサーバ31で受信され、クライアント制御部32に通知される。クライアント制御部32はサービス一覧取得部33を呼び出す。サービス一覧取得部33はディレクトリサービスクライアント39を利用してネットワーク上のディレクトリサービス装置4にアクセスし、ネットワーク上のアプリケーションサービスの一覧を取得し、その結果をXML生成部38のサービス一覧生成部を用いて、図2のようなXMLデータを生成する。このXMLデータは、利用者がいずれかのアプリケーションサービスを選択でき、さらに、選択されたアプリケーションサービスの識別子を中継サーバ3へ再度通知できるように構成されている。例えば、このXMLデータに記述される情報はアプリケーションサービスの名称と識別子であり、他にディレクトリサービスから取得できるアプリサービス情報として場所や機能情報など利用者に提供されるべき情報などを含む。図2に示した例ではアプリケーションサービスの識別子としてアプリケーションサービスのURLを利用しているが、中継サーバ3とアプリケーションサービス間のミドルウェアにより、アプリケーションサービスを特定できるものならばURLにかぎらない識別子が

利用可能である。また、この例では利用者の入力手段としてアプリケーションサービス毎のハイパーリンクを利用しているが、HTMLのCSS機能を使い、アプリケーションサービス名籍を利用者がリストボックス等から選択し、送信ボタンを押すことで選択されたアプリケーションサービスの識別子を中継サーバ3へ送るようHTMLを構成することも可能である。生成されたHTMLデータはクライアント制御部32からHTTPサーバ31を介して要求元のWebブラウザに返送される。なお、この実施例では、ネットワーク上のアプリケーションサービスの一覧を保持したディレクトリサービス装置4がネットワーク上に存在する構成を示しているが、前述のように、ディレクトリサービス装置4がない構成でも、ディレクトリサービスクライアント機能がネットワーク上の全アプリケーションサービスのDirectory Service Agentに直接アクセスする方法や、全アプリケーションサービスのDirectory Service Agentに直接アクセスする構成も可能である。

【0012】クライアント端末装置1のWebブラウザでは、このHTMLデータを受信すると、例えば図3に示したように表示する。次に、この表示されたHTMLデータから利用者が特定のアプリケーションサービス名籍を選択すると、選択されたアプリケーションサービスの識別子を含む、選択された情報がHTML変換されてHTTPで再度中継サーバ3へ送られ、クライアント制御部32に通知される。クライアント制御部32では受信した情報から選択されたアプリケーションサービスの識別子を抽出し、その結果をI/P仕様取得部34へ送り、該当するアプリケーションサービスのI/P仕様記述データに対応したHTMLデータを生成する。さらに、I/P仕様取得部34では、前述のように、識別子で指定されたアプリケーションサービスへサービス間で共通のコマンドを送信するか、ディレクトリサービス装置4に登録されている情報を取得するか、クライアント側モジュールであるproxyから対応するアプリケーションサービスのAPI仕様を取得するかして、I/P仕様記述データを取得する。要は、この際に取得できる仕様の記述方法がHTMLデータ以外で記述されており、I/P仕様取得部34が一定のルールに基づいてHTMLデータに変換することが本発明の主題である。I/P仕様記述はHTMLデータへの変換アルゴリズムが定数であるものであればどのような形式でも構わない。このようなI/P仕様記述の例として図4にCSSの記述の例を示す。Sun Microsystems社のJava言語では、各アプリケーションサービスのプログラムオブジェクトに、それ自身の持つ外部I/P関数の仕様を実行時に問い合わせることが可能で、アプリケーションサービスがJava環境で実装されている場合、アプリケーションサービス手段内に内蔵する各プログラムオブジェクトの外部I/P関数仕様をクライアント端末装置1からの要求時に収集し、動的に例えばXML形式のI/P仕様記述を生成することが可能

である。また、Java環境をベースとしたNet/INGミドルウェア環境では、中継サーバ内でクライアント側モジュールであるproxyから対応するアプリケーションサービスのAPI仕様をJavaのプログラムオブジェクトとして取得することが可能である。図5に、この場合のJava言語でのコード例を示す。この例ではI/P仕様記述を独自の形式で文字列データとして生成している。I/P仕様取得部34はI/P仕様記述を取得後、それをHTML生成部38のコマンド一覧生成部に渡す。コマンド一覧生成部はI/P仕様記述からコマンドを抽出し、各コマンド毎にコマンド引数入力項目生成部を呼び出してコマンド名とパラメータに対応する<FORM/>項目を生成する。HTML/HTML、NETなどミドルウェアがリソースを呼び出しの形式を実装している場合、コマンドは各外部I/P関数名、パラメータは各関数の引数採用される。

【0013】この場合の<FORM/>の構成例を図5に示す。ここでの構成例では、各パラメータに対応する<INPUT type="text" name="command">を配置する。コマンド名は<INPUT type="submit" name="command">のHTML属性としてセットする。また、<FORM/>の属性として送信ボタンが押下されたときに、コマンド名およびパラメータ値が送信される送信先URLとして中継サーバ3のHTTPサーバ31のURLを指定する。コマンド一覧生成部はコマンド毎にコマンド引数入力項目生成部を呼び出してこれらの各コマンドに対応する<FORM/>を生成し、それらを結合して一つのHTMLデータを生成し、I/P仕様取得部34に返送する。I/P仕様取得部34はこれをクライアント制御部32へ返送し、このHTMLデータはクライアント端末装置1のWebブラウザに送信される。図7に生成されるHTMLデータの例とWebブラウザ上での表示例を示す。Webブラウザ上で利用者が一つのコマンドに対し各パラメータを入力してコマンド名でラベルされたボタンを押下すると、クライアント端末装置1は図8に示すようなHTMLメッセージを送信する。中継サーバ3のHTTPサーバ31はこれを受け取り、アプリケーション識別子、コマンド名、パラメータ値をセットでクライアント制御部32へ送信する。クライアント制御部32ではこれをサービス起動制御部37へ送信する。サービス起動制御部37では、コマンド生成手段を駆動してアプリケーションサービスにアクセスするためのミドルウェアに対応した形で図9に示す変換を行い、コマンドを生成し、アプリケーション識別子で指定されるアプリケーションサービス宛てに送信する。これにより、そのアプリケーションサービスからコマンド実行結果(処理結果)が戻ってくると、サービス起動制御部37はその結果をコマンド実行結果解析部36に渡し、その返り値の型に応じたHTMLデータを生成し、そのHTMLデータをクライアント制御部32およびHTTPサーバ31を経由して当該クライアント端末装置1へ送り、Webブラウザに表示する。この際、コ

マンド実行結果がアプリケーション識別子であった場合には識別子をHTMLデータとして返すのではなく、この識別子をクライアント制御部32に渡し、クライアント制御部32では、1/x処理取得部34を起動して識別子に対応するアプリケーションサービスの1/x情報を取得し、そのHTMLデータを生成し、それをWebブラウザに送信する。

【0014】

【発明の効果】以上説明したように、本発明によれば、請求項1記載の発明では、HTTPサーバを含む中継サーバがクライアント端末装置からの要求情報を受信し、その要求情報に基づいてアプリケーションサービス手段に処理を依頼し、処理結果をそのアプリケーションサービス手段から受信してクライアント端末装置へ返送する際、クライアント端末装置からの要求情報中に指定されたアプリケーションサービスについてサービスに対する処理依頼方法に関する情報が中継サーバにより取得され、取得された情報に基づいてその中継サーバによりHTMLデータが生成され、クライアント端末装置へ返送されるので、ブラウザを利用したクライアントサーバシステムにおいて、ネットワーク上へ新たなアプリケーションサービスが追加された時点で、自動的に、クライアント端末装置からアプリケーションサービスが利用できるようになる。また、請求項2記載の発明では、請求項1記載の発明において、クライアント端末装置からの処理要求が処理内容に対応したコマンドを送信する形で行われる場合、中継サーバから送られてきた入力項目を用いてクライアント端末装置のブラウザ上で入力または選択された内容が中継サーバへ送られるので、アプリケーションサービスの各コマンドを利用者が直接指定して、新しいアプリケーションサービスの提供をクライアント端末装置から利用することができる。また、請求項3記載の発明では、請求項2記載の発明において、アプリケーションサービス手段が各コマンドの実行にコマンドの指定の他にパラメータの指定が必要な場合、中継サーバにより、ブラウザ上におけるパラメータ値の指定を可能にさせる入力項目をもつHTMLデータが生成されるので、パラメータの指定が必要な場合でも、新しいアプリケーションサービスの提供をクライアント端末装置から利用することができる。また、請求項4記載の発明では、請求項1、請求項2、または請求項3記載の発明において、ネットワーク上の全てのアプリケーションサービスに対する処理の依頼方法に関する情報がディレクトリサービス手段から取得されるので、その依頼方法の中継サーバに予め登録しておかなくても、新たなサービスが追加された時点で、中継サーバがそれらの情報を動的に収集できる。

【0015】また、請求項5記載の発明では、請求項1、請求項2、または請求項3記載の発明において、アプリケーションサービスに対する処理の依頼方法に関す

る情報が各アプリケーションサービス手段により管理され、各アプリケーションサービス間で共通な特定のコマンドを各アプリケーションサービス手段へ送ることにより前記依頼方法に関する情報が取得されるので、同時に、その依頼方法の中継サーバに予め登録しておかなくても、新たなサービスが追加された時点で、中継サーバがそれらの情報を動的に収集できる。また、請求項6記載の発明では、請求項1乃至請求項5のいずれかに記載の発明において、クライアント端末装置上で、送られてきた入力項目を含むHTMLデータがブラウザ上に表示され、利用者がその入力項目に入力値を入力するか入力値を選択するかすることにより要求内容がクライアント端末装置から中継サーバへ送られ、アプリケーションサービスに対する処理の依頼方法に関する情報にしたがって、中継サーバにより前記入力値からサービスに対する実際のコマンドおよびパラメータが生成され、アプリケーションサービス手段へ送信されるので、容易な操作で所望のアプリケーションサービス内容を指定してそのアプリケーションサービスを実行させることができる。また、請求項7記載の発明では、請求項8記載の発明において、中継サーバにより、コマンド送信に対する応答としてアプリケーションサービス手段から処理結果が受信され、その処理結果を元にHTMLデータが生成され、生成されたHTMLデータがクライアント端末装置へ返送されるので、容易な操作で所望のアプリケーションサービス内容を指定してそのアプリケーションサービスの結果をクライアント端末装置で得ることができる。また、請求項8記載の発明では、請求項7記載の発明において、アプリケーションサービス手段へのコマンド送信に対する処理結果としてアプリケーションサービス手段から当該アプリケーションサービスまたは別のアプリケーションサービスのサービス識別子が返送され、中継サーバにより、返送されたサービス識別子に対応するアプリケーションサービスに関する処理の依頼方法が取得され、取得された情報に基づきHTMLデータが生成され、生成されたHTMLデータがクライアント端末装置へ返送されるので、アプリケーションサービス手段による処理結果がそれと連携する別のアプリケーションサービスの処理と関連する場合、容易にこの別のアプリケーションサービスも受けられる。

【0016】また、請求項9記載の発明では、請求項1、請求項2、または請求項3記載の発明において、アプリケーションサービスが、サービスオブジェクトとして外部インターフェースから参照可能なドキュメントを用いて実装され、アプリケーションサービスに対する処理の依頼方法に関する情報としてそのサービスオブジェクトの外部インターフェースに関する情報がアプリケーションサービス手段から取得されるので、アプリケーションサービスがドキュメントを用いてサービスオブジェクトとして実装されている場合においても、請求項1、



【図6】本発明の一実施例を示すクライアント・サーバシステムの他の説明図である。

【図7】本発明の一定実施例を示すクライアント・サーバシステムの他の説明図である。

【図8】本発明の一実施例を示すクライアント・サーバシステムの他の説明図である。

【図9】本発明の一実施例を示すクライアント・サーバシステムのフロー図である。

70 1 754724 000000

**0**      **██████████**

31 ~~2004-4~~

3-3 サーズス一覽表

85 771 #4 2701 9/28

28 9 ? \*~7Z

1000

【図1】本発明の一実施例を示すクライアントサーバシステムのシステム構成図である。

【図2】本発明の一実施例を示すクライアントサーバシステムの説明図である。

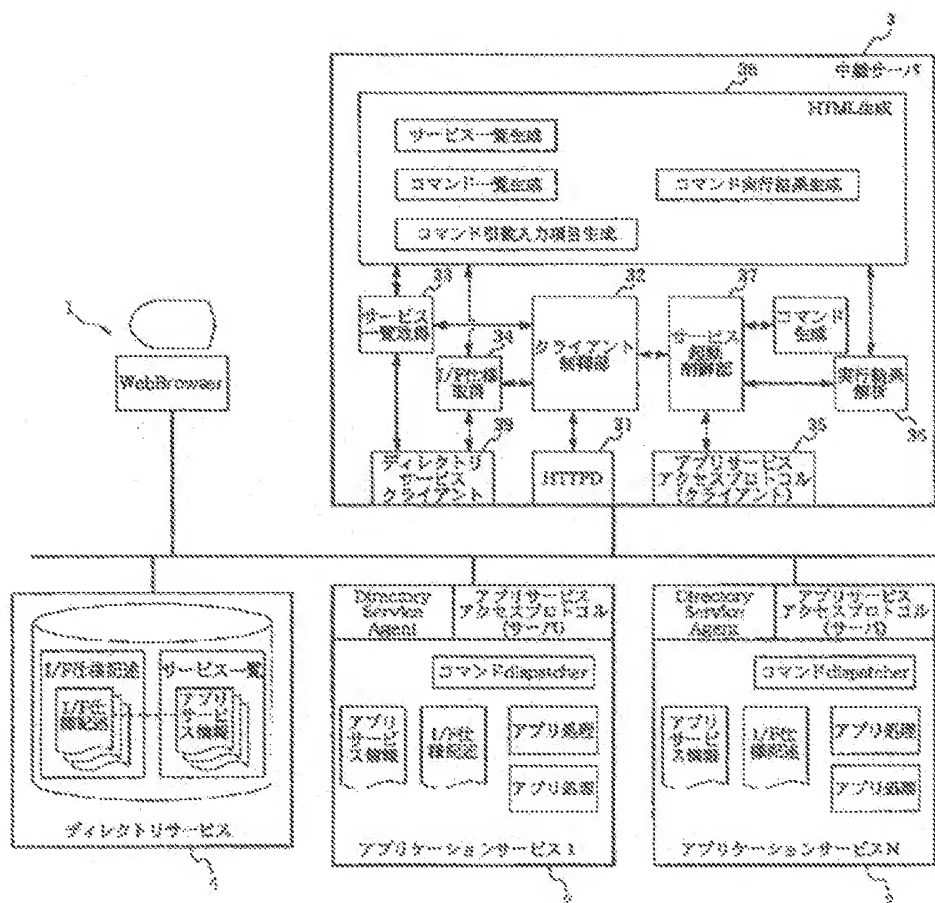
【図3】本発明の一実施例を示すクライアントサーバシステムの他の構成図である。

【図4】本発明の一実施例を示すクライアントサーバシステムの他の説明図である。

【図5】本発明の一実施例を示すクライアントサーバシステムの他の態様図である。

[illegible]

1111

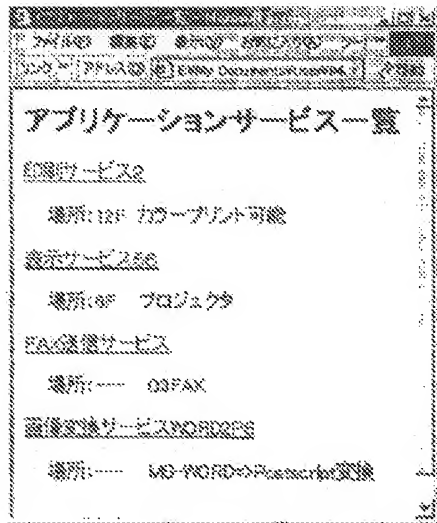


**Abstract**

```
module ApiService {
    interface print {
        typedef long jobId;
        typedef string urlID;
        jobId print(in string string_to_print);
        jobId printURL(in string url);
        urlID getControlId();
    }

    interface control {
        typedef long jobId;
        jobId currentJob();
        void cancelJob(in jobId jobId);
    }
}
```

【 図3 】



【 図5 】

```

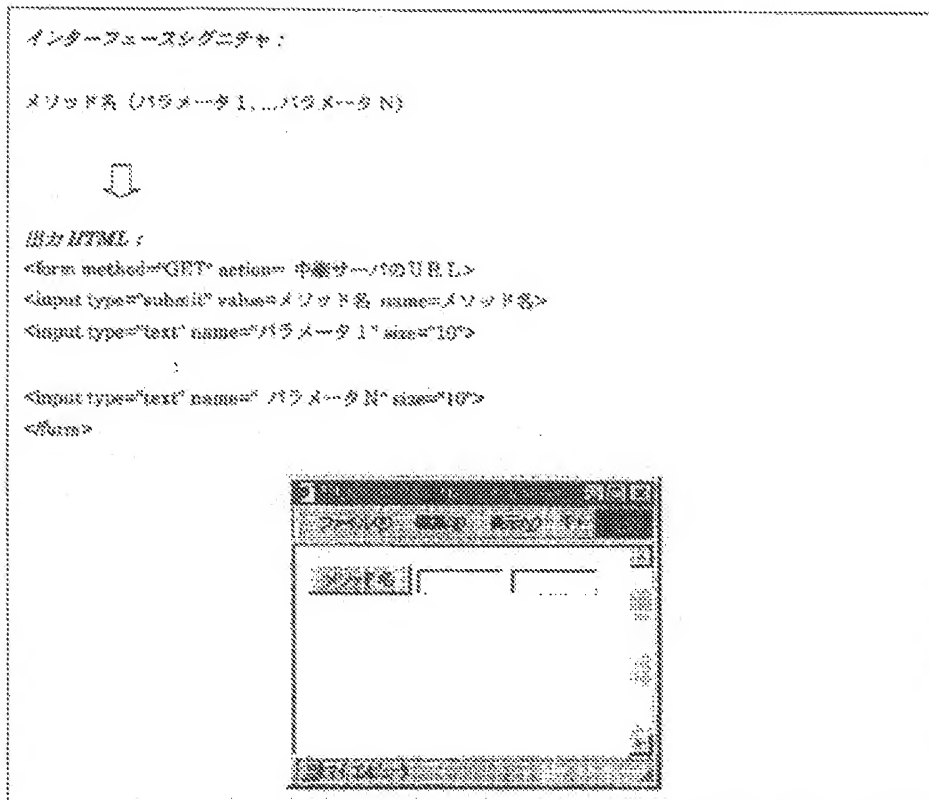
import java.lang.reflect.*;
import java.awt.*;

class SampleClass {

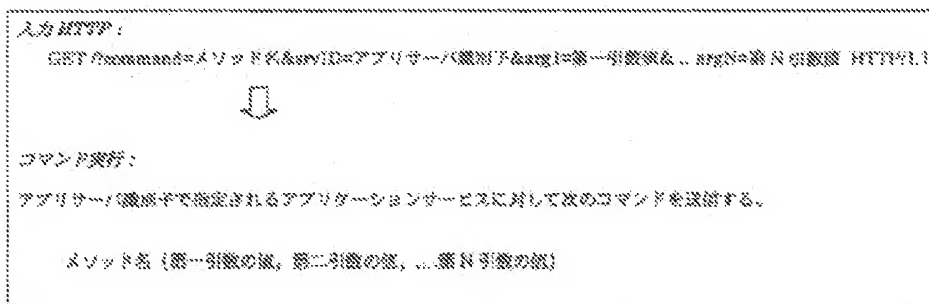
    String getPFDesc(Object proxy) {
        String out;
        Class c = proxy.getClass();
        Method[] methodArray = c.getMethods();
        for (int i = 0; i < methodArray.length; i++) {
            out = "name:" + methodArray[i].getName();
            out = out + " type:" + methodArray[i].getReturnType().getName();
            Class[] parameterTypes = methodArray[i].getParameterTypes();
            for (int k = 0; k < parameterTypes.length; k++) {
                out = out + " param:" + parameterTypes[k].getName();
            }
            return out;
        }
    }
}

```

【 図6 】



【 図8 】



【 図7 】

```

<html>
<body>
<h2>印刷サーバのコマンド</h2>

<form method="GET" action="http://gateway.xxx.co.jp/entry">
    <!-- action 属性に中継サーバのURL -->
    <input type="hidden" name="srvID" value="http%3A%2F%2FprintSRV.xxx.co.jp">
    <!-- hidden 項目としてアプリサーバの識別子 -->
    <input type="submit" value="print" name="command">
    <input type="text" name="arg1" size="10">
</form>

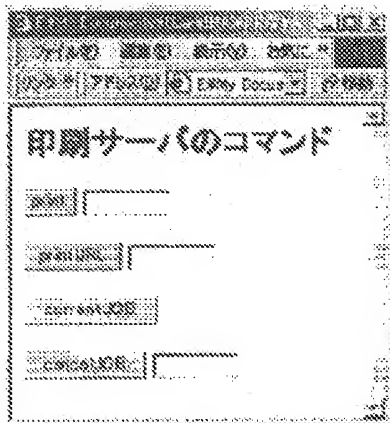
<form method="GET" action="http://gateway.xxx.co.jp/entry">
<input type="hidden" name="srvID" value="http%3A%2F%2FprintSRV.xxx.co.jp">
<input type="submit" value="printURL" name="command"> </form>

<form method="GET" action="http://gateway.xxx.co.jp/entry">
<input type="hidden" name="srvID" value="http%3A%2F%2FprintSRV.xxx.co.jp">
<input type="submit" value="currentJOB" name="command">
</form>

<form method="GET" action="http://gateway.xxx.co.jp/entry">
<input type="hidden" name="srvID" value="http%3A%2F%2FprintSRV.xxx.co.jp">
<input type="submit" value="cancelJOB" name="command">
<input type="text" name="arg1" size="10">
</form>

</body>
</html>

```





【 図9 】

